# Program Assignment
# Recursive Descent Parser

Compiler Theory #1

21600193

Hyo-Rim, Kim

May 13th, 2019

1. EBNF

program ⟶ stmt-sequence | ε

stmt-sequence ⟶ statement {; stmt-sequence}

statement ⟶for-stmt | while-stmt | if-stmt | class-stmt | func-stmt | assign-stmt | main-stmt | else-stmt

main-stmt ⟶ main(){ stmt-sequence }

if-stmt ⟶ if (expr) {stmt-sequence}
else-stmt ⟶ else {stmt-sequence} | if-stmt

class-stmt ⟶ class id {stmt-sequence}

func-stmt ⟶ id(func-parameter)

assign-stmt ⟶ id = expr | id++ | id- -

for-stmt ⟶ for(assign-stmt ; expr ; assign-stmt){stmt-sequence}

while-stmt ⟶ while(expr){stmt-sequence}

expr ⟶<term> {<addop> <term>} | <term> {<cmpop> <term>}

func-parameter ⟶ id {, id} | literal

term ⟶ <factor> {<mulop> <factor>}

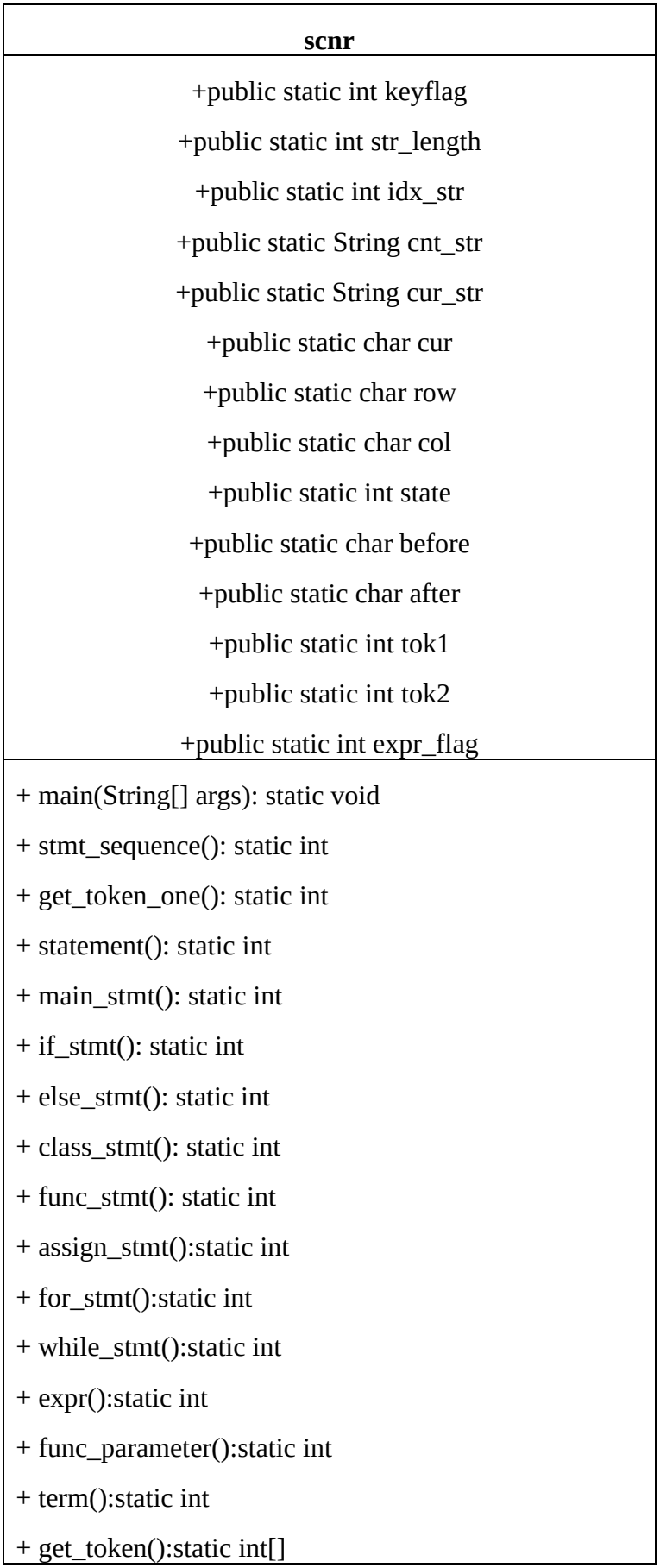
addop ⟶ + | -

cmpop ⟶ < | <= | > | >= | == | !=

mulop ⟶ * | /| %

factor ⟶ number | id | (expr)

2. Design Document
    1. UML



| |
|---|
| + readFile(final String fileName): static String<br>+ get_token_type(String str, int state): static int |

3. Result

1. Use Java

```
class : keyword
MyClass : id
{ : left curly brace
main : keyword
( : left parenthesis
) : right parenthesis
{ : left curly brace
int : keyword
$_Time0 : id
= : assignment symbol
22 : number literal
; : semicolon
if : keyword
( : left parenthesis
$_Time0 : id
< : greater than symbol
10 : number literal
) : right parenthesis
{ : left curly brace
out.println : keyword
( : left parenthesis
" : double quote symbol
Good : literal
morning. : literal
" : double quote symbol
) : right parenthesis
; : semicolon
} : right curly brace
else : keyword
if : keyword
( : left parenthesis
$_Time0 : id
< : greater than symbol
20 : number literal
) : right parenthesis
{ : left curly brace
out.println : keyword
( : left parenthesis
" : double quote symbol
Good : literal
day. : literal
" : double quote symbol
) : right parenthesis
; : semicolon
} : right curly brace
else : keyword
{ : left curly brace
out.println : keyword
( : left parenthesis
" : double quote symbol
Good : literal
evening. : literal
" : double quote symbol
) : right parenthesis
; : semicolon
} : right curly brace
} : right curly brace
} : right curly brace
Parsing Ok
hyorm@ariselab:~/cmpr/hw2_21600193/use_javac/src$
```

2. Use Ant

```
build:

run:
     [java] class : keyword
     [java] MyClass : id
     [java] { : left curly brace
     [java] main : keyword
     [java] ( : left parenthesis
     [java] ) : right parenthesis
     [java] { : left curly brace
     [java] int : keyword
     [java] $_Time0 : id
     [java] = : assignment symbol
     [java] 22 : number literal
     [java] ; : semicolon
     [java] if : keyword
     [java] ( : left parenthesis
     [java] $_Time0 : id
     [java] < : greater than symbol
     [java] 10 : number literal
     [java] ) : right parenthesis
     [java] { : left curly brace
     [java] out.println : keyword
     [java] ( : left parenthesis
     [java] " : double quote symbol
     [java] Good : literal
     [java] morning. : literal
     [java] " : double quote symbol
     [java] ) : right parenthesis
     [java] ; : semicolon
     [java] } : right curly brace
     [java] else : keyword
     [java] if : keyword
     [java] ( : left parenthesis
     [java] $_Time0 : id
     [java] < : greater than symbol
     [java] 20 : number literal
     [java] ) : right parenthesis
     [java] { : left curly brace
     [java] out.println : keyword
     [java] ( : left parenthesis
     [java] " : double quote symbol
     [java] Good : literal
     [java] day. : literal
     [java] " : double quote symbol
     [java] ) : right parenthesis
     [java] ; : semicolon
     [java] } : right curly brace
     [java] else : keyword
     [java] { : left curly brace
     [java] out.println : keyword
     [java] ( : left parenthesis
     [java] " : double quote symbol
     [java] Good : literal
     [java] evening. : literal
     [java] " : double quote symbol
     [java] ) : right parenthesis
     [java] ; : semicolon
     [java] } : right curly brace
     [java] } : right curly brace
     [java] } : right curly brace
     [java] Parsing Ok

BUILD SUCCESSFUL
Total time: 1 second
```

5. User Manual
    1. Use ant (directory name)
        1. ant version
            - Apache Ant(TM) version 1.10.5 compiled on March 28 2019
    2. build.xml

```
<project name="scnr_prsr" default="build" basedir=".">

    <property name="src" value="src"/>
    <property name="build" value="build"/>
    <property name="doc" value="doc"/>

    <path id="lib.path">
        <pathelement location="${build}" />
    </path>

    <target name="init">
        <mkdir dir="${build}"/>
    </target>
    <target name="build" depends="init">
        <javac srcdir="${src}" destdir="${build}" debug="true"
includeantruntime="false">
        </javac>
    </target>

    <target name="run" depends="build">
        <java classname="scnr_prsr" fork="true" dir="." maxmemory="4096m">
            <classpath location="."/>
            <classpath refid="lib.path"/>
            <arg file="data/test.txt"/>
        </java>
    </target>

    <target name="clean">
        <delete dir="${build}"/>
```

```
        </target>
</project>
```

6. command
    - ant build
    - ant run
        - this build.xml already set the file name(test.txt)
7. Use Javac (directory name)
    1. java version
        - openjdk version "11.0.6" 2020-01-14
        - OpenJDK Runtime Environment (build 11.0.6+10-post-Ubuntu-1ubuntu118.04.1)
        - OpenJDK 64-Bit Server VM (build 11.0.6+10-post-Ubuntu-1ubuntu118.04.1, mixed mode)
    2. command
        - javac scnr.java
        - java scnr [file name]